

# Курс лекций “Знакомство с ПЛИС”

Лекция №1. Что умеют ПЛИС и  
зачем они нужны

**Лектор**

Толкачев Максим

руководитель группы FPGA НТЦ Метротек

[m.tolkachev@metrotek.ru](mailto:m.tolkachev@metrotek.ru)

## Содержание курса

- Курс для новичков в ПЛИС.
- Желателен опыт программирования или embedded разработки.
- Презентации выкладываются после лекции.
- Список рекомендованного ПО:
  - Windows версия: Quartus Prime Lite Edition 19.1:  
[Download Center for FPGAs](#)
  - Windows версия: ModelSim-Intel FPGA Edition  
(вместе с Quartus)

## Формат

- 4 лекции (1 лекция = 1ч);
- 20 минут ответов на вопросы после каждой лекции;
- вопросы можно присылать на почту или задавать на стриме.

11.05 Пн.	12.05 Вт.	13.05 Ср.	14.05 Чт.	15.05 Пт.	16.05 Сб.	17.05 Вс.
18:00 Лекция 1		18:00 Лекция 2				
18.05 Пн.	19.05 Вт.	20.05 Ср.	21.05 Чт.	22.05 Пт.	23.05 Сб.	24.05 Вс.
18:00 Лекция 3		18:00 Лекция 4				

- **Лекция №1:** Что умеют ПЛИС, и зачем они нужны.
  - Что это такое, где используется и из чего состоит.
- **Лекция №2:** Языки описания аппаратуры. Часть 1.
  - Знакомимся с HDL/HLS языками. Начинаем изучать SystemVerilog. Запускаем симулятор.
- **Лекция №3:** Языки описания аппаратуры. Часть 2.
  - Углубляемся в SystemVerilog.
- **Лекция №4:** Путь проекта для ПЛИС.
  - Используя полученные знания, собираем свой первый проект для ПЛИС.

- Курс **вводный** и многое **упрощено** для простоты понимания.
- Курс **не подготовит** к позиции Junior FPGA, но станет отправной точкой подготовки.
- В курсе не рассматриваются вопросы:
  - Производства ПЛИС;
  - Аппаратной реализации ПЛИС на уровне полупроводниковых элементов.
- Все слайды, видео и файлы проектов буду выложены в открытый доступ.

# Лекция №1. Что умеют ПЛИС, и зачем они нужны

- I. Терминология;
- II. Что такое ПЛИС;
- III. Где ПЛИС применяются;
- IV. Производители ПЛИС;
- V. Архитектура ПЛИС:
  - a. общая структура;
  - b. функции блоков;
- VI. Советы для самостоятельной работы.

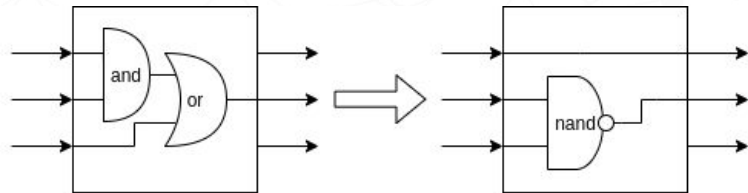


# I. Терминология

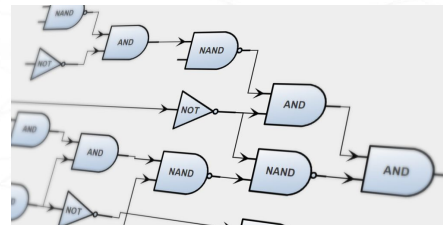
- **ПЛИС** (Программируемая Логическая Интегральная Схема) ~=  
**FPGA** (Field-Programmable Gate Array) ~=  
**CPLD** (Complex Programmable Logic Device)
- **ASIC** (Application-Specific Integrated Circuit)

## II. Что такое ПЛИС

- Программируемая



- Логическая



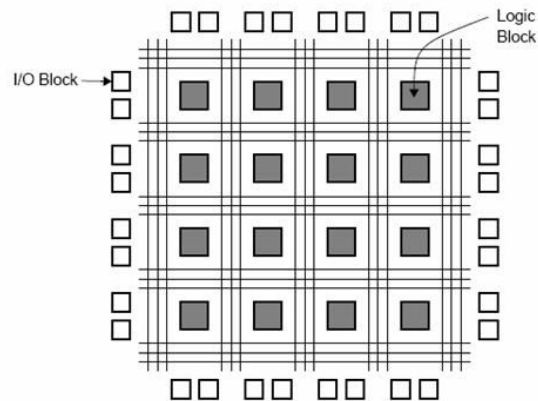
- Интегральная  
Схема



- **Field-Programmable** →



- **Gate Array** →



# ПЛИС не исполняет программы!

- Программа

```
int main() {
    int x=10,y=15;
    return 0;
}
```

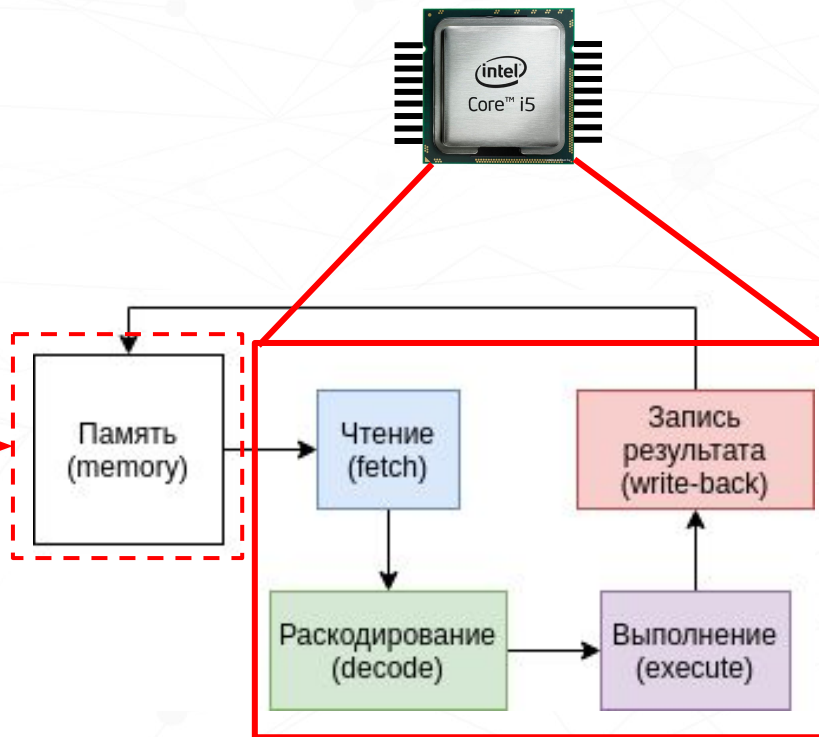


0000000000000000 <main>:

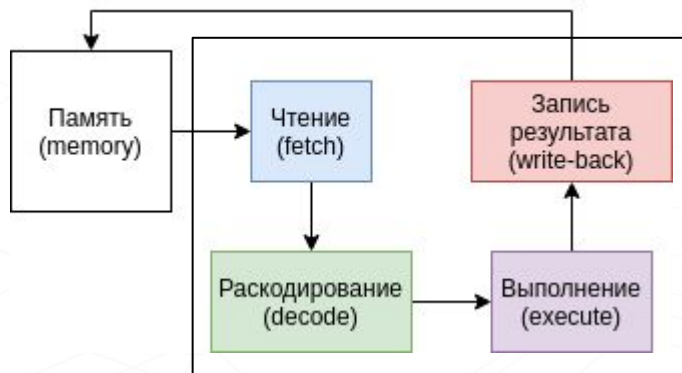
```
0: 55          push    %rbp
1: 48 89 e5     mov     %rsp,%rbp
4: c7 45 fc 0f 00 00 00 movl    $0xf,0x4(%rbp)
b: b8 00 00 00 00 mov     $0x0,%eax
12: 5d          pop     %rbp
17: c3          retq
```



- Упрощенный конвейер процессора



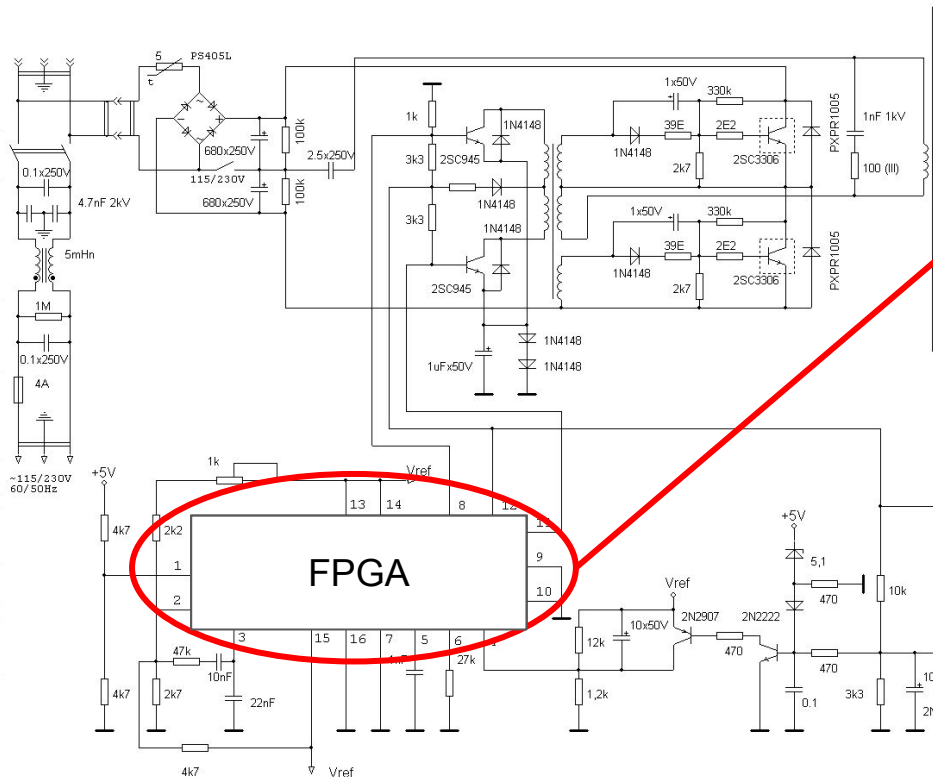
# Как работает схема на примере процессора



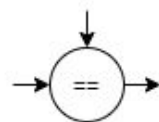
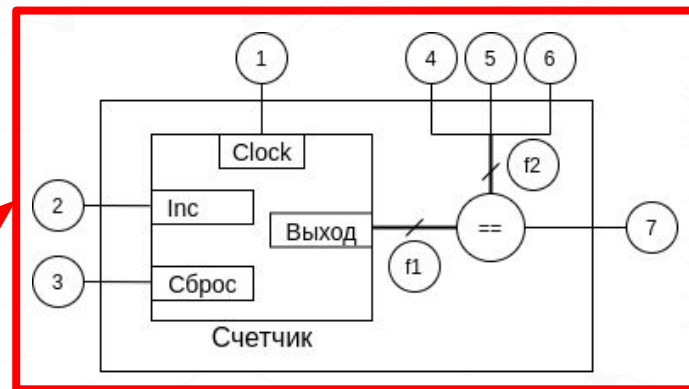
№ цикла \ Стадии	1	2	3	4	5	6	7
Чтение (fetch)	И1	И2	И3	И4	И5	И6	И7
Раскодирование (decode)		И1	И2	И3	И4	И5	И6
Выполнение (execute)			И1	И2	И3	И4	И5
Запись результата (write-back)				И1	И2	И3	И4

- Блоки внутри работают одновременно (параллельно)
- Для работы блоков с памятью (регистров) нужен периодический сигнал (clock)
- Частота этого сигнала напрямую влияет на производительность

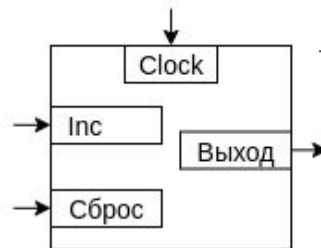
- Схемотехника:



- Цифровая схемотехника:

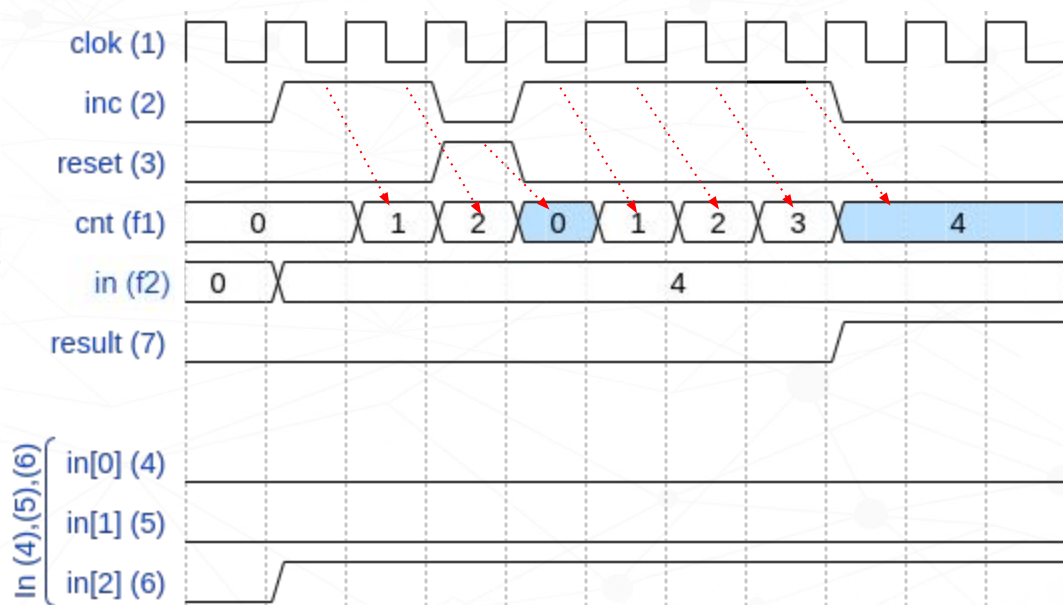
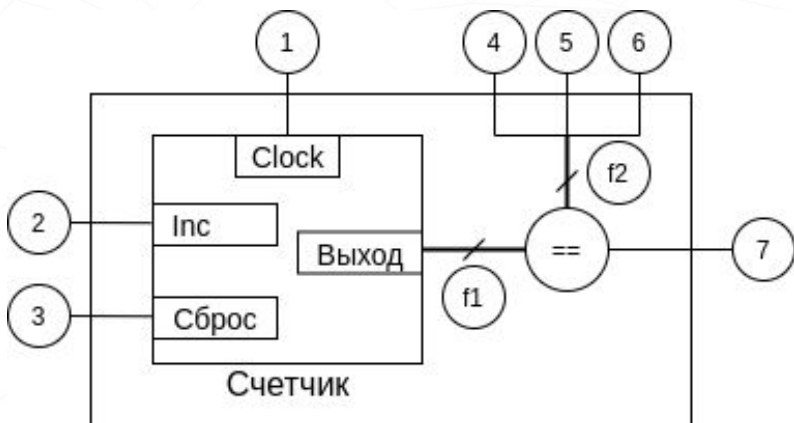


-- **Компаратор.** Сравнение входов.  
На выходе "1" если входы  
равны. "0" если не равны.



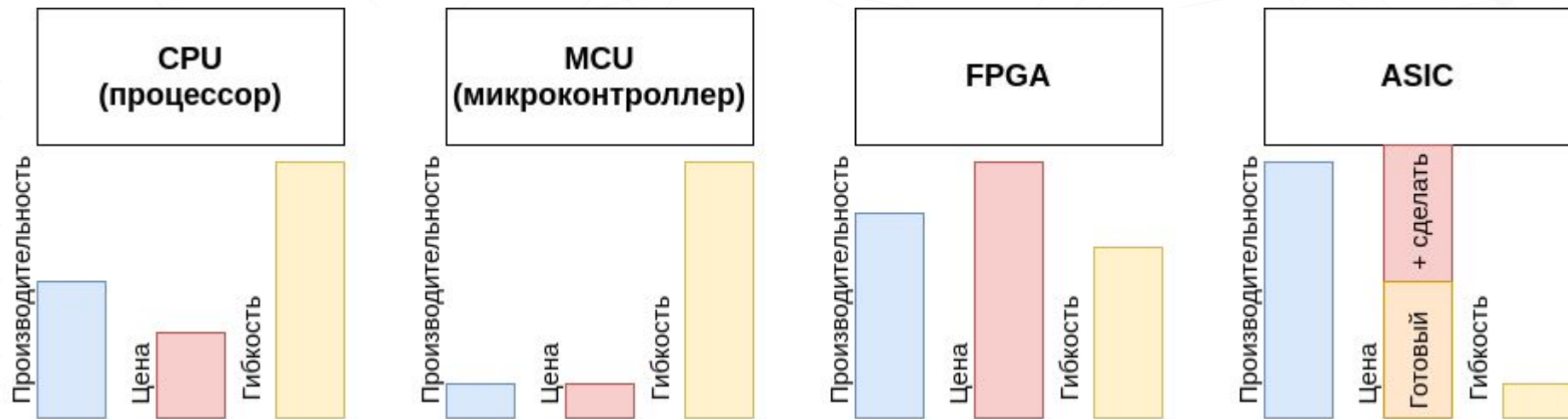
- **Счетчик.** Считает количество тактов, когда был “1” на входе “Inc.”
- “1” на входе “Сброс” сбрасывает счетчик.





- На входе микросхемы цифровые сигналы (0,1)
- Абстрагируемся от физической реализации
- Основные элементы для работы:
  - Регистры (элементы памяти)
  - Логические вентили (И, ИЛИ, НЕ ...)
- Все элементы работают одновременно и уже есть в ПЛИС. Мы их только соединяем в нужном порядке.





ПЛИС используют, когда производительности CPU не хватает и готовых специальных схем для решения задачи нет.

### III. Где ПЛИС применяются

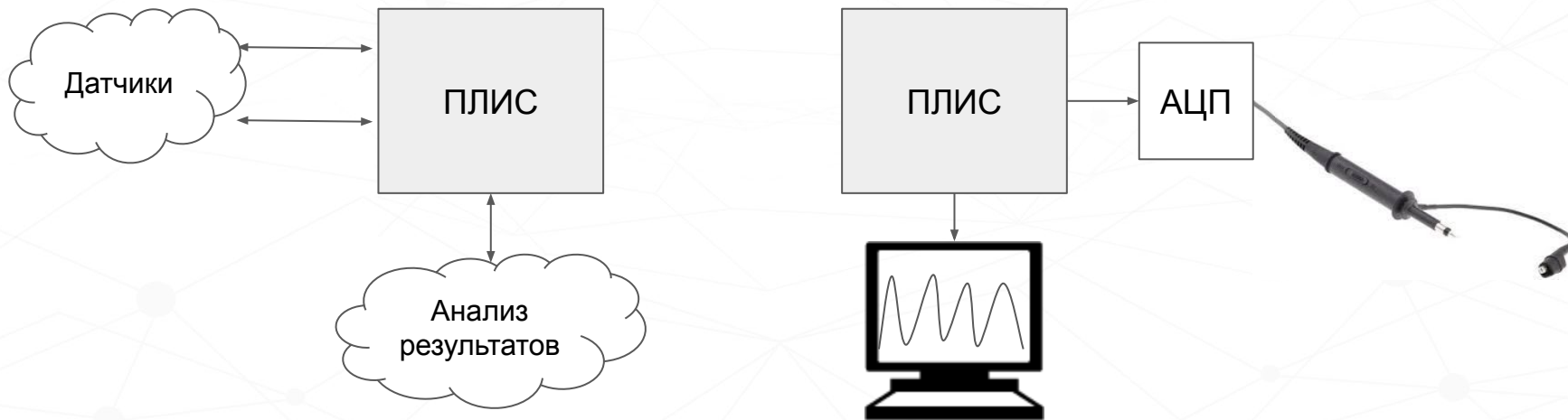
## By Xilinx:

- Video and Image Processing
- Computational Storage
- Database and Data Analytics
- Financial Technology
- High Performance Computing
- Network Acceleration

## By Intel FPGA:

- Automotive
- Embedded Vision
- Test and Measurement
- Broadcast
- Military, Aerospace, Government
- Wireless

## 1. Контрольно-измерительная техника



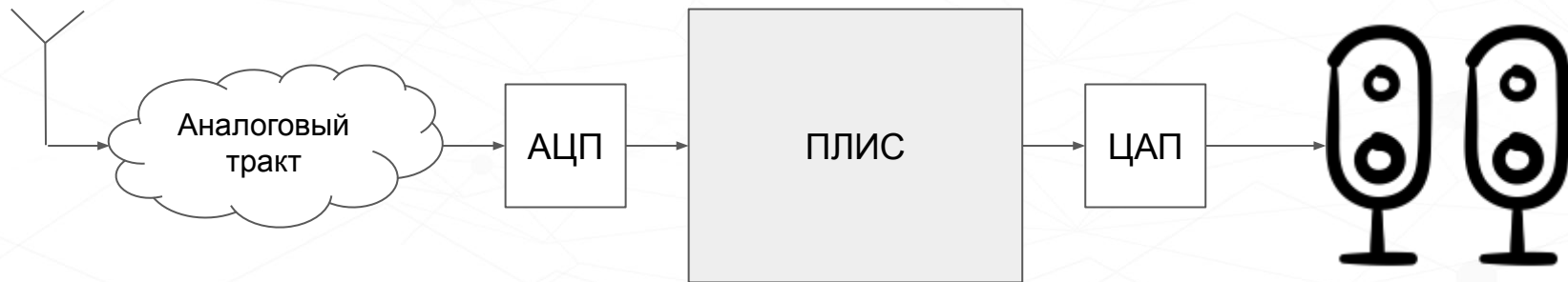
Точные измерения, детерминированные по времени воздействия на данные с датчиков предобработка большого потока данных.

## 2. Преобразователь интерфейсов



ПЛИС могут обрабатывать ЛЮБОЙ цифровой интерфейс и любое количество таких интерфейсов.

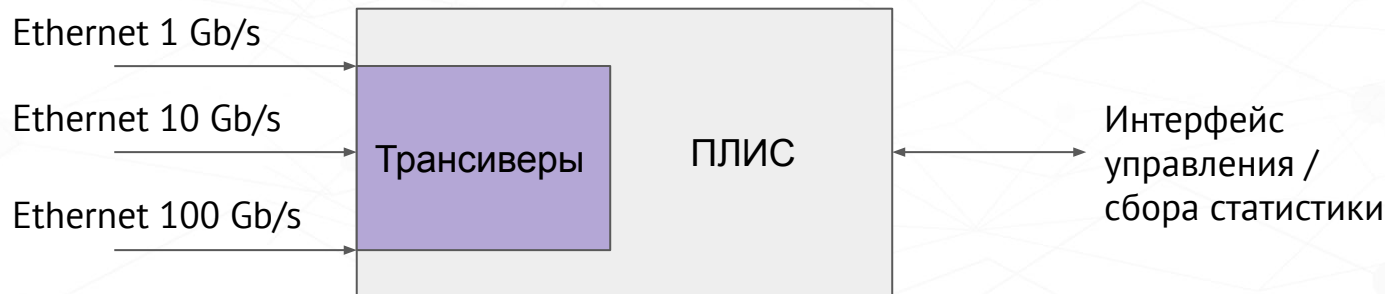
### 3. Цифровая обработка сигналов (ЦОС)



На архитектуру ПЛИС отлично ложатся различные цифровые фильтры, преобразования Фурье, демодуляцию и пр.

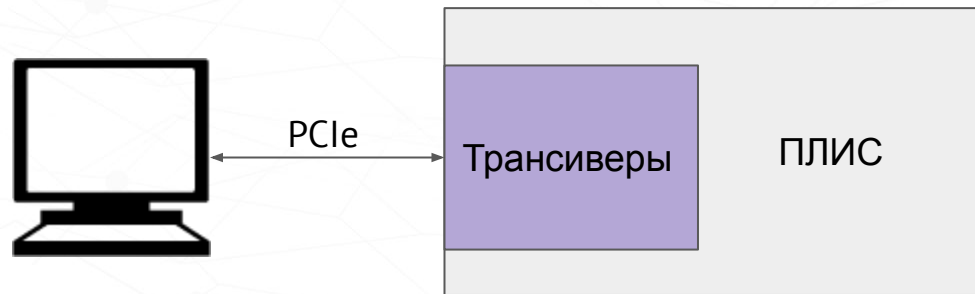


## 4. Быстрая и точная обработка сетевых пакетов



Обработка сетевых пакетов: генерация, прием, измерение задержки, точные временные метки приема пакетов, балансировки трафика и т. д. И все со 100% нагрузкой и без потерь.

## 5. Специальные ускорители

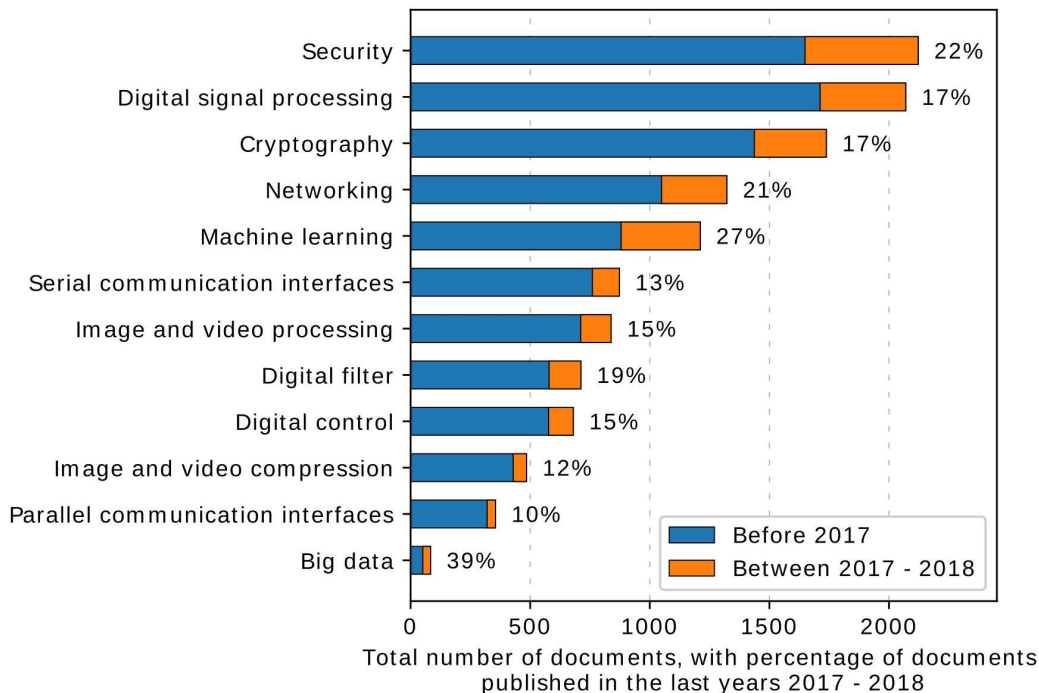


Большие ПЛИС с высокоскоростными трансиверами. Альтернатива GPU для другого класса задач или для уменьшения энергопотребления.

FPGA top applications categories in the research community from 1992 to 2018

Области научных работ по  
ПЛИС (FPGA):

- безопасность;
- ЦОС;
- криптография;
- сети;
- машинное обучение;
- последовательные интерфейсы;
- обработка видео и изображений;
- ...



Field Programmable Gate Array Applications -- A Scientometric  
Review (2019): <https://www.mdpi.com/2079-3197/7/4/63>

## IV. Производители ПЛИС

- Xilinx
- Intel FPGA
- Lattice
- Microsemi
- QuickLogic
- Gowin
- ВЗПП-С (ВОРОНЕЖСКИЙ ЗАВОД ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ-СБОРКА)
- EFINIX
- Cologne Chip

Доли рынка ПЛИС по вендорам

**QuickLogic**

1,0%

**Microsemi**

4,0%

**Lattice**

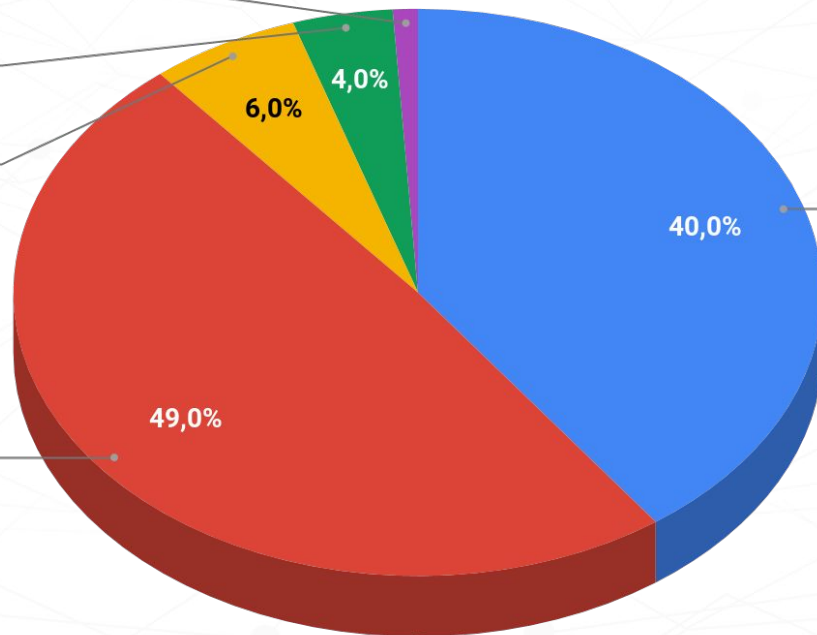
6,0%

**Xilinx**

49,0%

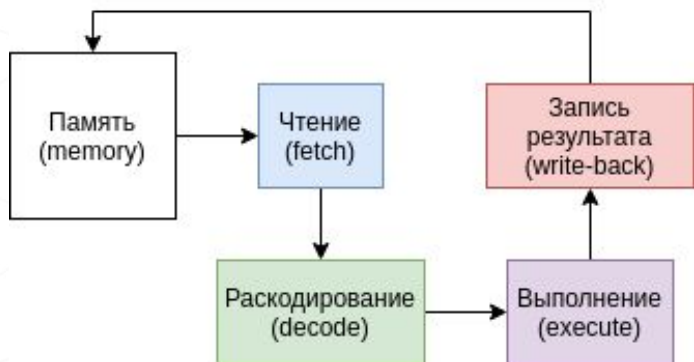
**Intel (Altera)**

40,0%

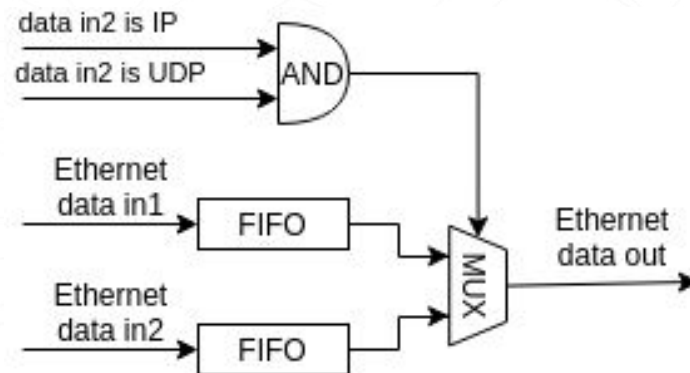


## V. Архитектура ПЛИС

## 1. Блок схема:

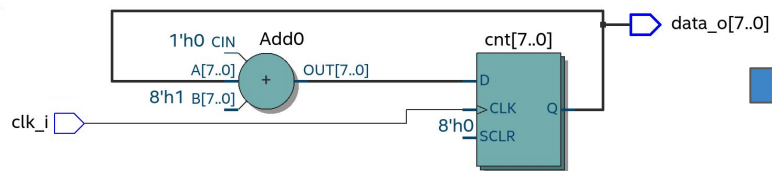


## 2. Структурная схема:

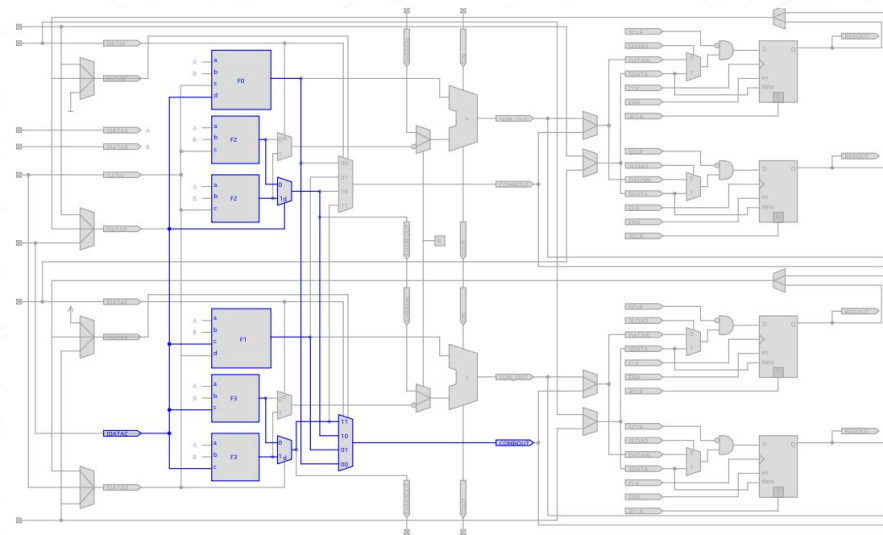




## 3. Уровень регистровых передач (RTL):



## 4. Вентильный уровень (gate):



## I/O буферы:

- Clock buffers -- соединяют входы со специальными линиями тактирования
- IO buffers -- согласование с внешней схемой.

## Intel: ALM, Xilinx: CLB:

- Ячейки с логикой и регистрами

## Intel: M9K/M10K, Xilinx: BlockRAM:

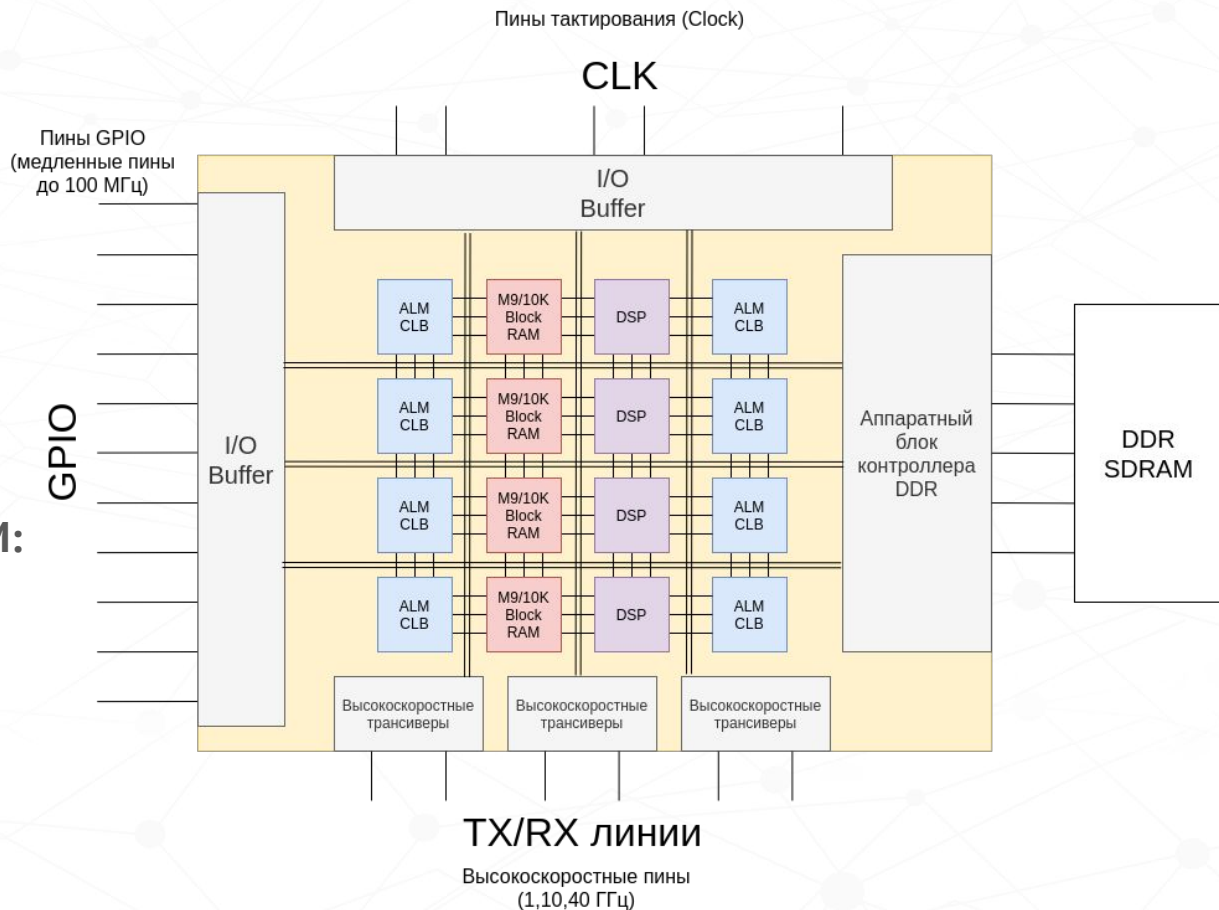
- Внутренняя память

## DSP:

- Аппаратные умножители

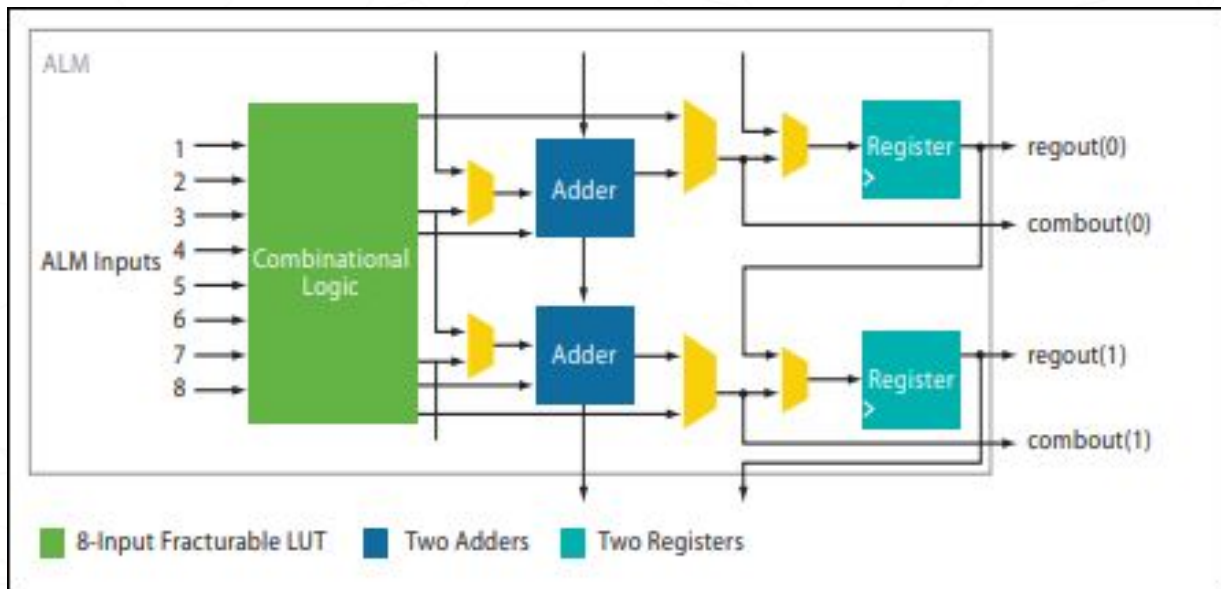
## Трансиверы:

- Сериализация/десериализация



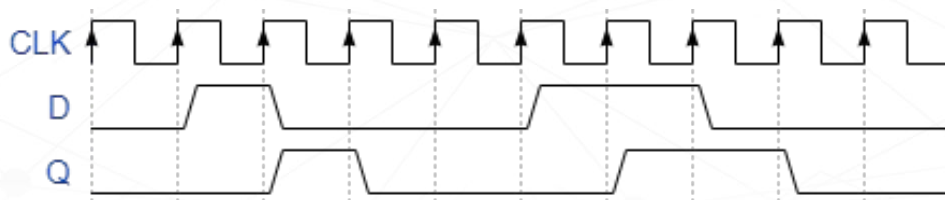
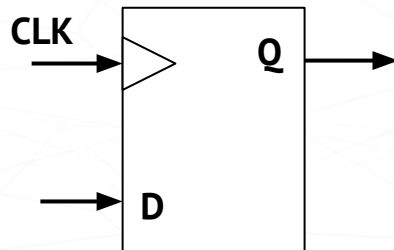
# Что внутри ПЛИС. Логические блоки

- ALM или CLB -- главные внутренние блоки обработки сигналов.  
ALM (Adaptive Logic Module) CLB (Configurable Logic Block)



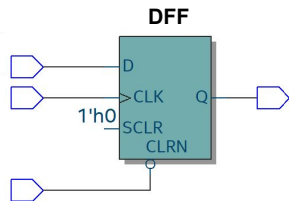
- LUT = таблица истинности.  
**Комбинационная** схема.
- Register = триггер = D-триггер. Ячейка памяти.  
**Последовательная** схема.

- D-триггер:



# Что внутри ПЛИС. Регистр

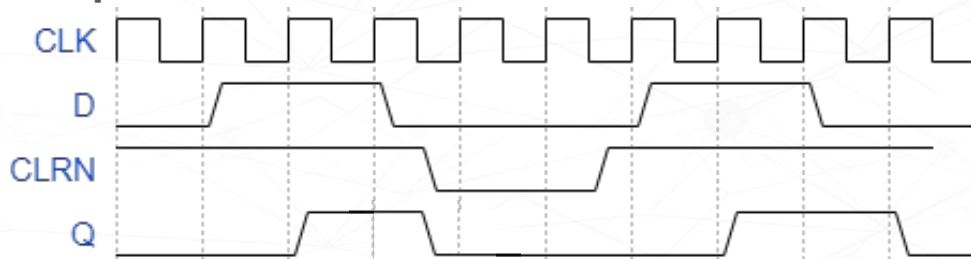
- Регистр: триггер, D-триггер, D flip-flop, DFF:



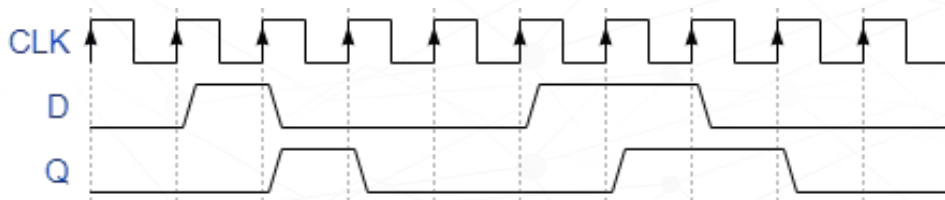
Inputs			Output
CLR N	CLK	D	Q
0	X	X	0
1	⌋	0	0
1	⌋	1	1
1	0	X	Q <sub>o</sub>
1	1	X	Q <sub>o</sub>

**Q<sub>o</sub> (Q old):** Старое значение **Q**

**Сброс:**

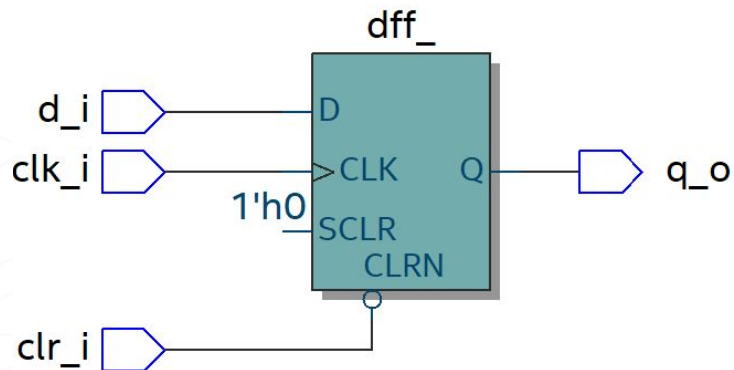


**CLR N = 1. Рабочее состояние:**

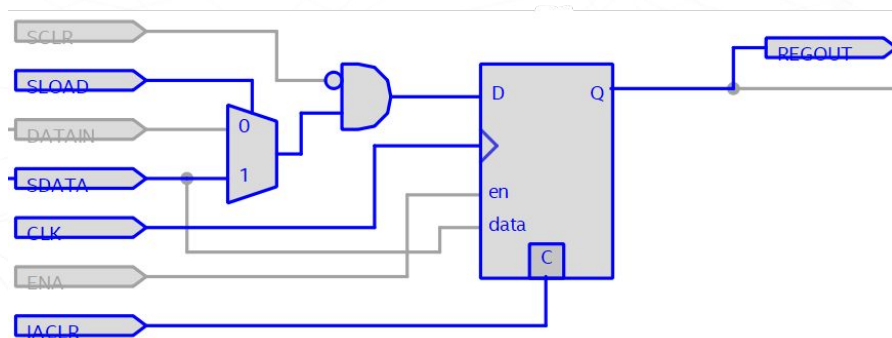


- Регистр: триггер, D-триггер, D flip-flop, DFF:

RTL:

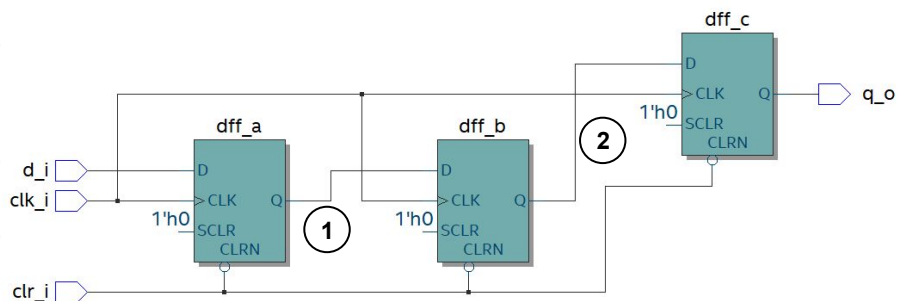


Gate level:

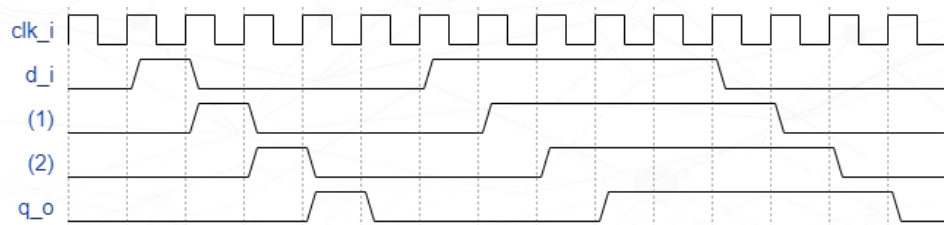


# Что внутри ПЛИС. Регистр

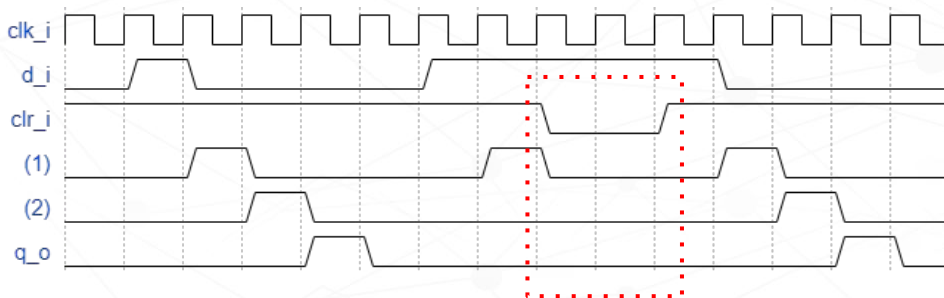
- Сдвиговый регистр:



`clr_i = 1`. Рабочее состояние:

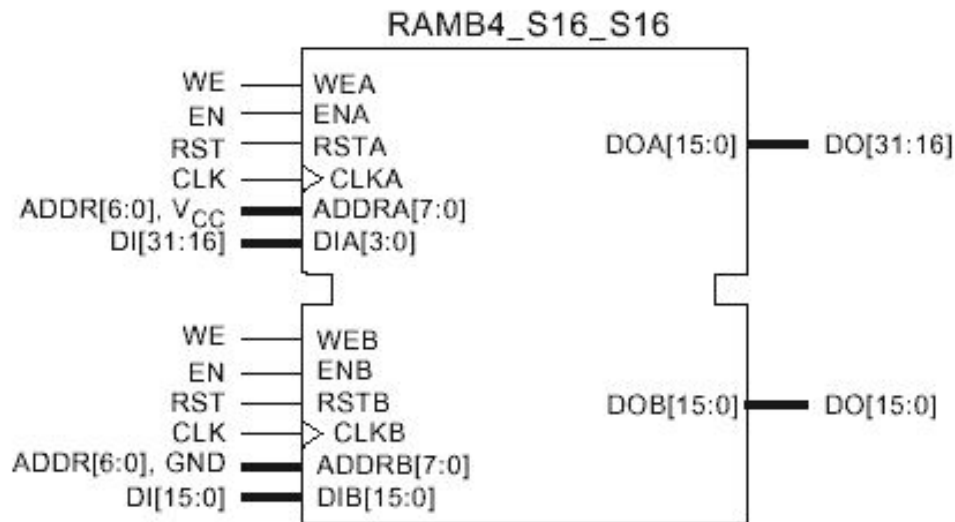


Сброс:



# Что внутри ПЛИС. Блочная память

- М9К/М10К ... или BlockRAM -- блоки внутренней памяти. НЕ РЕГИСТРЫ.



- 9К, 10К ... -- размер блоков:
  - М9К = 8 192 бита
  - М10К = 10 240 бит
- BlockRAM = 32 768 бит

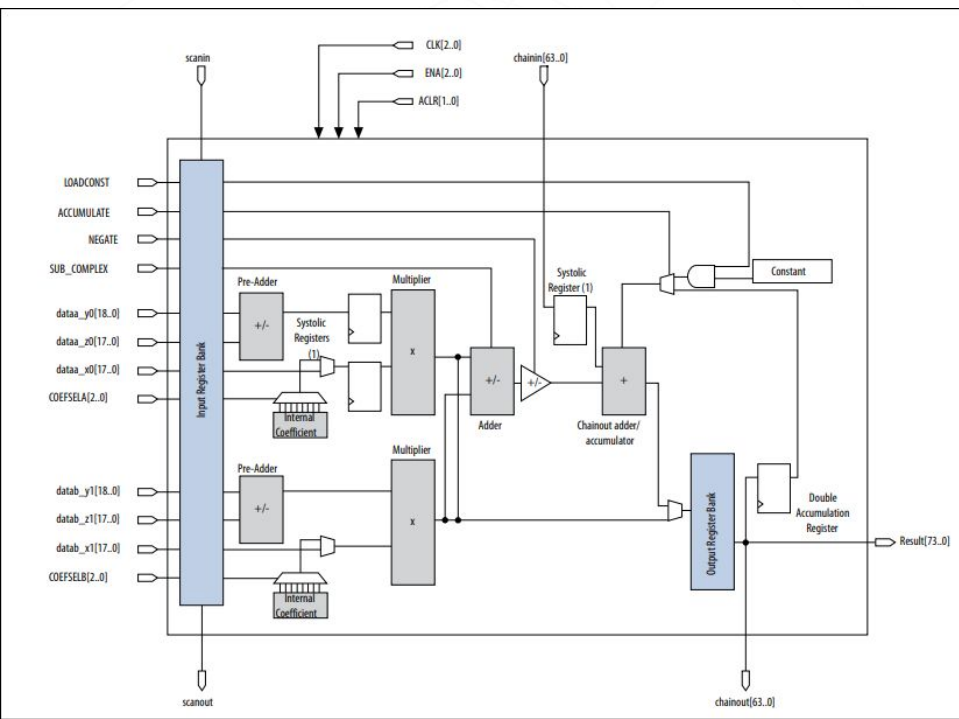
Количество блоков (штук):

- Xilinx 7 серия: от 135 до 2176
- Cyclone V: от 176 до 1220
- Xilinx UltraScale Plus: до 3008

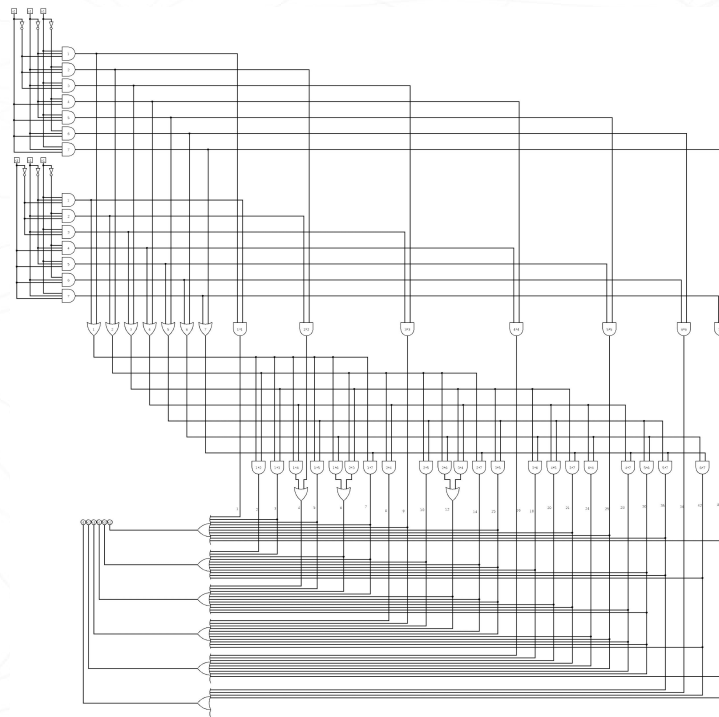


- Специальные умножители -- потому что умножение на логике это сложно.

Аппаратный умножитель 18x18:



Умножитель 3x3 на логике:



- Синтезатор тактовой частоты: PLL
- Трансиверы (SERDES)
- Аппаратные блоки (Hard-IP):
  - SDRAM контроллеры для доступа к памяти DDR2|3|4
  - Ethernet MAC контроллеры (10G и больше)
  - Процессор (ARM Cortex A9): HPS | PS
  - ...

Поколение Семейство	V	10
Cyclone	<p><b>Есть</b> с ядром <b>ARM</b>. Маленький размер и энергопотребление.</p> <p><b>Трансиверы:</b> до 6.144 Гб/с (до 12 штук)</p> <p><b>ALM:</b> от 9 000 до 113 000</p>	<p><b>10 GX</b></p> <p><b>Нет</b> ядра <b>ARM</b>. Развитие Cyclone V.</p> <p><b>Трансиверы:</b> до 12.5 Гб/с (до 12 штук)</p> <p><b>ALM:</b> от 31 000 до 80 000</p>
		<p><b>10 LP</b></p> <p><b>Нет</b> ядра <b>ARM</b>. Перерождение старого семейства Cyclone III.</p> <p><b>Трансиверы:</b> нет</p> <p><b>ALM</b> (пересчет из LE): от 2 000 до 43 000</p>
Arria	<p><b>Есть</b> с ядром <b>ARM</b>. Средняя линейка.</p> <p><b>Трансиверы:</b> до 6.144 Гб/с (до 36 штук)</p> <p><b>ALM:</b> от 28 000 до 190 000</p>	<p><b>Есть</b> с ядром <b>ARM</b>. Средняя линейка.</p> <p><b>Трансиверы:</b> до 25.78 Гб/с (до 6 штук)</p> <p><b>ALM:</b> от 61 000 до 427 000</p>
Stratix	<p><b>Нет</b> ядра <b>ARM</b>. Максимальная производительность в поколении.</p> <p><b>Трансиверы:</b> до 12.5 Гб/с (до 66 штук)</p> <p><b>ALM:</b> от 128 000 до 360 000</p>	<p><b>Есть</b> с ядром <b>ARM</b>. Максимальная производительность.</p> <p><b>Трансиверы:</b> до 28.3 Гб/с (до 96 штук)</p> <p><b>ALM</b> (пересчет из LE): от 137 000 до 3 720 000</p> <p><b>Hyperflex</b></p>

Поколение Семейство	7-series	UltraScale	UltraScale Plus
Zynq	<p><b>С ядром ARM.</b> Низкая цена и производительность. <b>Трансиверы:</b> до 12.5 Гб/с (до 16 штук) <b>LC:</b> от 23 000 до 350 000</p>	--	<p><b>С ядром ARM.</b> Высокая производительность. <b>Трансиверы:</b> до 33 Гб/с (до 16 штук) <b>LC:</b> от 103 000 до 930 000</p>
Artix / Spartan	<p><b>Нет ядра ARM.</b> Маленькие и средние ПЛИС. <b>Трансиверы:</b> до 6.6 Гб/с (до 16 штук) <b>LC:</b> от 6 000 до 215 000</p>	--	--
Kintex / Virtex	<p><b>Нет ядра ARM.</b> Широкая линейка. <b>Трансиверы:</b> до 28.5 Гб/с (до 16 штук) до 13.1 Гб/с (до 80 штук) <b>LC:</b> от 65 000 до 876 000</p>	<p><b>Нет ядра ARM.</b> Большая производительность. <b>Трансиверы:</b> до 30.5 Гб/с (до 60 штук) <b>CLB:</b> от 145 000 до 2 500 000</p>	<p><b>Нет ядра ARM.</b> Максимальная производительность. <b>Трансиверы:</b> до 58 Гб/с (до 48 штук) до 32.7 Гб/с (до 128 штук) <b>CLB:</b> от 325 000 до 4 000 000</p>

## **VI. Советы для самостоятельной работы**

1. Начать читать книгу: “Цифровая схемотехника и архитектура компьютера” Дэвид М. Харрис и Сара Л. Харрис.
2. Лекции “FPGA начального уровня” от FPGA-Systems.ru:  
<https://youtu.be/xcjR6BpTqhM>
3. Выбрать для себя язык описания аппаратуры для изучения (будет в Лекции №2).
4. Научиться работать с симулятором.
5. Выбрать вендор первой отладочной платы ПЛИС: лучше Xilinx или Intel
  - доступность,
  - интерфейсы для “развлечений”.
5. Начать знакомиться со средой разработки:
  - Xilinx: Vivado
  - Intel: Quartus

6. Начать “просматривать” стандарт выбранного языка
7. Углубиться в симуляцию:
  - автоматические тесты без анализа “времянок” глазами.
8. Углубиться в написание кода:
  - прочесть design guidelines от вендора
  - изучить <http://www.sunburst-design.com/> для best practice
  - прочесть “Advanced FPGA Design: Architecture, Implementation, and Optimization” Steve Kilts
9. Взяться за “серьезную” задачу

**Для новичков:**

- Статья от Ивана Шевчука: [Junior FPGA Design Engineer: как стать? / Хабр](#)
- [Цифровая схемотехника и архитектура компьютера- второе издание](#) Дэвид М. Харрис и Сара Л. Харрис
- Новости мира FPGA и не только: [FPGA-Systems - Сообщество FPGA разработчиков](#)
- Примеры кода для DIY проектов: [fpga4fun.com - where FPGAs are fun](#)

**Обязательно к прочтению для развития:**

- Рекомендации к написанию HDL кода от Inte: [Intel Quartus Prime Pro Edition User Guide: Design Recommendations](#)
- Рекомендации к написанию HDL кода от Xilinx: [Xilinx Synthesis and Simulation Design Guide \(UG626\)](#)
- Стандарт SystemVerilog: [1800-2017 - IEEE Standard for SystemVerilog--Unified Hardware Design, Specification, and Verification Language - IEEE Standard](#)
- Best Practice: [Sunburst Design World Class Verilog, SystemVerilog & UVM Verification training. Classes include expert and advanced Verilog, Verilog Synthesis SystemVerilog and UVM Training classes.](#)



## Материалы для профессиональной разработки:

- Документация по конкретным семействам ПЛИС. Примеры:
  - Xilinx 7 серия UG:
    - CLB: [7 Series FPGAs Configurable Logic Block User Guide \(UG474\)](#)
    - RAM: [7 Series FPGAs Memory Resources User Guide](#)
    - Прошивка: [7 Series FPGAs Configuration User Guide \(UG470\)](#)
  - Intel Cyclone V UG:
    - [Cyclone V Device Handbook Volume 1: Device Interfaces and Integration](#)
- Советы по улучшению HDL кода: [Advanced FPGA Design: Architecture, Implementation, and Optimization: Steve Kiltz: 9780470054376](#)
- Советы по использованию SystemVerilog для верификации: [SystemVerilog for Verification: A Guide to Learning the Testbench Language Features](#)